

What is claimed is:

1. A method of generating a patch file from an old version of data consisting of a series of elements and a new version of data consisting of a series of elements where both the old and new versions of data are stored in a memory and accessible by a data processor, the method comprising the steps of:

establishing an alphabet comprising a plurality of letters for processing the old and new versions of data wherein a word consists of a series of elements represented by one or more letters of the alphabet;

sorting the old version of data with the data processor alphabetically according to the established alphabet to create a first sorted list of words and maintaining a pointer for each element of the first sorted list of words indicating the element's original location in the old version of data;

sorting the new version of data with the data processor alphabetically according to the established alphabet to create a second sorted list of words and maintaining a pointer for each element of the second sorted list of words indicating the element's original location in the new version of data;

recursively comparing the first and second sorted lists of words one word at a time for a match of the words;

upon finding a match of the words, searching the first and second sorted lists of words again to find the largest sequence of coinciding elements preceding and succeeding the match of words;

storing the largest sequence of coinciding elements in a coincidences list;

processing the coincidences list to remove duplicative coincidences, creating a refined list of coincidences;

sorting the refined list of coincidences by pointer in the new version data, creating a sorted list of coincidences;

adding the pointer of each coincidence in the new version data to the largest sequence of coinciding elements, creating an end block list;

creating a patch file from the sorted coincidences list and the end block list; and

compressing the patch file into a secure, portable compressed archive for distribution.

2. The method of claim 1 wherein the patch file includes a series of write commands.

3. The method of claim 2 wherein creating a patch file from the sorted coincidences list and end block list includes the steps of:

(a) finding a largest block of coinciding elements of the sorted coincidences list that matches the second sorted list of words and recording the location of the largest found block;

(b) in the area before the previously found block, finding the next largest block of coinciding elements of the sorted coincidences list that matches the second sorted list of words and recording the location of the next largest found block;

(c) repeating step (b) until

(i) the first member in the sorted list of coincidences is reached; or

(ii) there are no matches between the second sorted list of words and the sorted coincidences list;

(d) writing a write from the old version of data command and pointer and length information to the patch file if the first member of the sorted list of coincidences is reached; and

(e) writing a write from patch file command and length and patch information to the patch file when there is no match between the second sorted list of data and the sorted coincidences list.

4. The method of claim 1 wherein one or more patch files are compressed into the .ZIP file format to decrease the storage and transfer requirements of the patch files.

5. The method of claim 1 wherein the .ZIP file further includes encryption and authentication using digital signatures to secure the contents of the patch files from unauthorized access and to validate the identity of the creator of the archive.

6. The method of claim 1 wherein the .ZIP file is in the format of a self-extracting .ZIP file.

7. The method of claim 6 wherein the self-extracting .ZIP file includes a rules-based form of intelligence to detect the presence of the appropriate files to be patched and to determine how the patching process should proceed.

8. A system for generating a patch file from an old version of data consisting of a series of elements and a new version of data consisting of a series of elements, the system comprising:

a data processor;

a memory storing the old and new versions of data;

means for sorting the old version of data with the data processor alphabetically according to an established alphabet to create a first sorted list of data and for maintaining a pointer for each element of the first sorted list of data indicating the element's original location in the old version of data;

means for sorting the new version of data with the data processor alphabetically according to an established alphabet to create a second sorted list of data and for maintaining a pointer for each element of the second sorted list of data indicating the element's original location in the new version of data;

means for searching the first and second sorted lists of data to find a match of data;

means for storing the match of data as a sequence of coinciding elements in a coincidences list;

upon finding a match of data, means for searching the first and second sorted lists of data again to find the largest sequence of coinciding elements preceding and succeeding the match of data;

means for storing the largest sequence of coinciding elements in the coincidences list;

means for processing the coincidences list to remove duplicative coincidences, creating a refined list of coincidences;

means for sorting the refined list of coincidences by pointer in the new version of data, creating a sorted list of coincidences;

means for adding the pointer of each coincidence in the new version of data to the largest sequence of coinciding elements, creating an end block list;

means for creating a patch file from the sorted coincidences list and end block list; and

means for compressing the patch file into a secure, portable compressed archive for distribution.

9. The system of claim 8 wherein the means for creating a patch file includes:

means for finding the largest block of coinciding elements of the sorted coincidences list that matches the second sorted list of data and recording the location of the largest found block of coinciding elements in a memory;

means for finding the next largest block of coinciding elements of the sorted coincidences list that matches the second sorted list of data and recording the location of the next largest block of coinciding elements in the memory;

means for writing a write form the old version of data command and pointer and length information to the patch file; and

means for writing a write from patch file command and length and patch information to the patch file.

10. The system of claim 8 wherein one or more patch files are compressed into the .ZIP file format to decrease the storage and transfer requirements of the patch files.

11. The system of claim 8 wherein the .ZIP file further includes encryption and authentication using digital signatures to secure the contents of the patch files from unauthorized access and to validate the identity of the creator of the archive.

12. The system of claim 8 wherein the .ZIP file is in the format of a self-extracting .ZIP file.

13. The system of claim 12 wherein the self-extracting .ZIP file includes a rules-based form of intelligence to detect the presence of the appropriate files to be patched and to determine how the patching process should proceed.

14. A method of creating one or more patch files and aggregating the patch files into a secure, portable compressed archive for distribution.

15. The method of claim 14 wherein the archive includes one or more patch files in a compressed form to decrease the storage and transfer requirements of the patch files.

16. The method of claim 14 wherein the archive further includes encryption and authentication using digital signatures to secure the contents of the patch files from unauthorized access and to validate the identity of the creator of the archive.

17. The method of claim 14 wherein the archive is in the format of a self-extracting .ZIP file.

18. The method of claim 17 wherein the self-extracting .ZIP file includes a rules-based form of intelligence to detect the presence of the appropriate files to be patched and to determine how the patching process should proceed.

19. A system for managing and manipulating patch files comprising:
a program for compressing, extracting, encrypting and decrypting patch files into a secure, portable compressed archive.

20. The system of claim 19 wherein the program provides the ability to digitally sign the patch files as they are compressed.